






## Method and device for distributing program package for hierarchical software

**Publication number:** CN1264077 (A)  
**Publication date:** 2000-08-23  
**Inventor(s):** AIBORT ALUN [US]; BUREISEN [US]; SHUMAH DAIL [US] +  
**Applicant(s):** PLATINUM TECHNOLOGY KNOWLEDGE [US] +  
**Classification:**  
- **International:** G06F13/00; G06F15/16; G06F9/445; H04L29/06; H04L29/08;  
G06F13/00; G06F15/16; G06F9/445; H04L29/06; H04L29/08;  
(IPC1-7): G06F9/44  
- **European:** H04L29/06; H04L29/08N33  
**Application number:** CN19991026668 19991223  
**Priority number(s):** US19980220724 19981224

### Also published as:

 CN1191525 (C)  
 EP1014652 (A2)  
 EP1014652 (A3)  
 EP1014652 (B1)  
 US6938075 (B1)

more >>

Abstract not available for CN 1264077 (A)  
Abstract of corresponding document: EP 1014652 (A2)

A software distribution system for software bundles comprising a plurality of packages, at least two of which are required by a different set of nodes on a network. Rather than send all of the packages constituting the software bundle to each node, only the packages in the software bundle which are required by nodes below a certain link in the network are sent along that link, whereby to minimize the bandwidth required for transmitting the software bundle.

---

Data supplied from the *espacenet* database — Worldwide

[51] Int. Cl<sup>7</sup>

G06F 9/44

## [12] 发明专利申请公开说明书

[21] 申请号 99126668.4

[43]公开日 2000年8月23日

[11]公开号 CN 1264077A

[22]申请日 1999.12.23 [21]申请号 99126668.4

### [30] 优先级

[32]1998. 12. 24US [33]US [31]09/220,724

**[71]申请人 白金技术知识产权公司**

地址 美国纽约

[72]发明人 阿伦·艾博特 布雷特·彼得森  
戴尔·舒马赫

[74] 专利代理机构 中国商标专利事务所

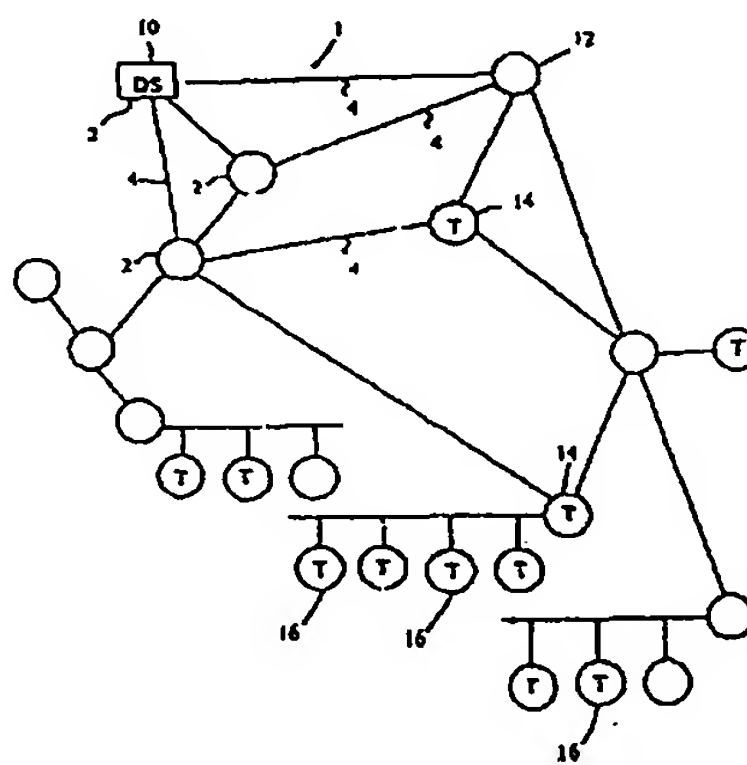
代理人 徐小琴

权利要求书 2 页 说明书 8 页 附图页数 5 页

**[54]发明名称** 用于分级软件分布程序包的方法与装置

### [57]摘要

一种用于集束软件的软件分布系统,包括若干程序包,该程序包的至少二个被一个不同的在网络上的节点组所要求。仅仅沿链发送被网络中某个链以下的节点所需要的软件组中的程序包,而不是将构成软件组的所有程序包发送到每个节点。由此,使得用于传送软件的带宽为最小。



ISSN 10.08-4274

知识产权出版社出版

## 权 利 要 求 书

1、用于分级软件分布的装置,允许分配包括至少第一和第二程序包到多个目标节点的软件包,其特征在于,所述装置包括:

用于传递软件包的一分布节点;

5 与所述分布节点联络的第一分枝节点, 设置第一分枝节点以接受从所述分布节点来的所述软件包;

第一和第二目标节点, 所述第一目标节点通过第一网络链与所述第一分枝节点联络, 而所述第二目标节点通过第二网络链与所述第二分枝节点联络; 所述第一程序包已在第二目标节点上呈现;

10 设置所述第一分枝节点以传送所述遍及所述第一网络链软件程序包, 而所述第二程序包仅遍及所述第二网络链;

因而第一和第二程序包二者被传送到所述第一和第二目标节点上。

2、根据权利要求 1 的装置, 其特征在于, 其中所述软件包作为连接的程序包遍及所述第一网络链而发送。

15 3、根据权利要求 1 的装置, 其特征在于, 其中, 提供给所述第一分枝节点有关程序包应被发送到目标节点去的信息。

20 4、根据权利要求 1 的装置, 其特征在于, 其中所述第一分枝节点通过第二分枝节点与所述第一目标节点联络, 所述第二分枝节点通过第一网络链与所述第一分枝节点联络, 而所述第二分枝节点通过第三网络链与所述第一目标节点联络; 所述第二分枝节点通过第四网络链进一步与第三目标节点联络。

25 5、根据权利要求 4 的装置, 其特征在于, 其中所述分枝节点每一个提供有关目标节点的信息, 每个分枝节点负责发送所述程序包到该目标节点和所述节点需要所述第一和第二程序包的该目标节点的信息; 而其中每个分枝节点沿每个分枝将信息发送到后来的节点, 整理每个分枝的所述信息以便仅包括通过那个分枝的目标节点。

30 6、根据权利要求 4 的装置, 其特征在于, 其中所述枝节点的每一个提供有关需要每一个所述第一和第二程序包的目标节点的信息, 并进一步提供有关所述分枝节点负责从所述分布节点发送到目标节点的信息, 同时分枝节点使用直接分枝以达到它所负责的所述每一个目标节点的信息, 由此, 每一个分枝节点能确定沿每一个直接分枝传送的程序包。

7、根据权利要求 1 的装置，其特征在于，程序包至少包括二个子程序包，同时其中所述二个子程序包在所述每一个目标节点上的安装必须以一种指定的顺序进行；其中

5 一个所述子程序包已安装出现在所述第一目标节点上；同时其中所述子程序包的二者被分配到所述第一目标节点上，同时没有一个子程序包被发送到所述第二目标节点上。

8、用于分级软件分布的装置，允许传送至少包括第一和第二程序包到若干目标节点上的软件包，其特征在于，所述装置包括：

传送软件包的一分布节点；

10 至少一个分枝节点与所述分布节点联络；以及第一和第二目标节点，所述第一目标节点通过第一网络链与所述分枝节点联络，而所述第二目标节点通过第二网络链与所述分枝节点联络，所述第一程序包已呈现在所述第二目标节点上。

15 设置每一个分枝节点以便独立地从所述分布节点接受所述第一和第二程序包的每一个；

设置所述分枝节点以便通过所述第一和第二网络链将所述第一程序包传送到所述第一和第二目标节点上，而所述第二程序包通过第二网络链传送到所述第二目标节点上；以及

设置目标节点以便一旦接受了程序包就安装每个程序包；

20 因而，完整的软件包被安装在每一个所述第一和第二目标节点上。

9、一种遍及至少一个共用的网络链分布软件包到至少第一和第二目标节点的方法，所述软件包包括至少第一程序包和第二程序包，所述第一软件程序包已经呈现在所述第二节点上，所述方法包括以下步骤，

25 遍及所述共同的网络链发送所述软件包；此后，仅发送所述第二程序包到所述第二目标节点；并发送所述软件包到所述第一目标节点；

因而，传送第一和第二程序包二者到所述第一和第二目标节点上。

## 用于分级软件分布程序包的方法与装置

5 本发明涉及软件分布且特别涉及软件经网络分布的方法，该方法使得用于传送软件的总带宽为最小。

众所周知，传送软件包是通过网络从服务器传送到需要程序包的一个或几个用户，通常，程序包的较新版本将被发行并需要传送给用户以便更新软件的老版本。

10 根据本发明，要传送的软件为几组文件，可能附带脚本，脚本在安装时运行，同时可能例如是在安装文件之前运行的安装前脚本或者之后运行的安装后脚本。

在某些情况下，给运行旧软件的所有系统升级并不是必须履行的责任或者对服务器说不可能具有运行软件的所有系统的记录。在  
15 这些情况中，运行软件的用户可以定期地向具有或不具有软件新版本的服务器查询，并且如果服务器通知用户其具有新版本时要求将软件送到用户，此方法已被商业软件分配器所采用，该分配器可以在整个网络上对其产品进行升级。

在许多情况中，上述软件传送的方法是不可接受的，例如，如  
20 果所述软件从来没有在用户系统中安装过，用户就无法了解它应该要求新的软件。此外，软件的升级通常必须立即安装以保持整个系统的完整性。考虑到这些情况，和其它类似的情况，提供从服务器到一个或多个用户主动地传送软件包的软件传送，而不是等待用户要求软件，就成为必需的。这种系统的一个例子是白金技术公司  
25 (Platinum Technology, Inc.)的 Auto Xfer 产品。

软件分布程序包一般是分级的，在软件分布程序包的那些文件中可以逻辑地分成更小的程序包，这些程序包对某些客户是需要的但对其它客户则不是。这些包含一组文件的程序包看成是简单的程序包，同时可能例如包括构成单一应用的文件。一个软件包也可以  
30 包含一个或几个组合的不包含实际文件程序包，而取代包含查询二





给其下一代节点的每一个。该下一代节点是下面节点所需要的并且在树状中包括那个下一代节点。一旦伴随它的任何程序包已经被打开，构成软件包的每一个较小程序包就可以被打开。

5 本发明进一步提供一种通过以树状型式个别地递送较小程序包的每一个而在网络两端由较小的程序包构成的软件包仅传送到节点上的方法与装置。在该节点上需要所述及的较小程序包，在该方法中传送节点是根，而需要较小程序包的用户是树状中的某些节点。一旦伴随它的任何程序包已经被打开，构成软件包的较小程序包就可以被打开。

10 在本发明的一种情况中，构成软件包的程序包作为单一的整体被递送，同时当所有进一步沿树状特殊分枝下降的节点不需要所有构成在要递送的分枝节点处接受的软件包的程序包时，在树状的分枝节点处进行再封装，然后一个再封装的向下经过分枝的每一个的软件程序包仅包含那个分枝以下的节点所需要的程序包。

15 在本发明的另一情况中，构成一个软件包的不同程序包，在它们所需要的目标节点处，独立地多点传送(multicasted)，然后一旦已经接受在目标节点处需要的所有程序包，就打开每个程序包。

20 本发明的这些和其它目的从本说明书的其余部分将会很明显。

图 1 表示在其上可以实现本发明的网络的一个例子。

图 2 表示图 1 所示经过网络的生成树的一个例子。

图 3 表示根据本发明的第一例的组合程序包的结构。

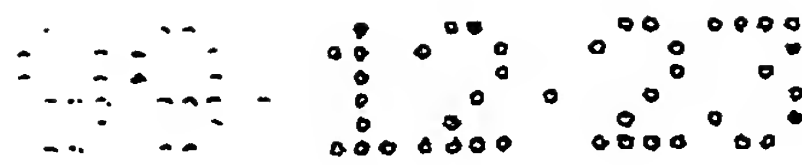
图 4A 表示根据本发明的一个例子的组合程序包的结构。

25 图 4B 表示图 3 所示的组合程序包的分布的第一例。

图 4C 表示图 3 所示的组合程序包的分布的第二例。

图 5 表示由第一实施例产生的一个阵列，在建立该阵列中程序包从图 4C 中的节点 Q 沿每一个链而传送。

图 6A 和图 6B 表示在本发明的第二实施例实现的不同的生成树。



参照图 1—3 在下文将说明本发明的第一实施例。

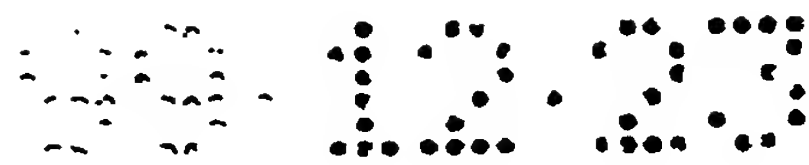
图 1 表示可以在其上使用本发明的网络的一个例子，这样一个网络具有节点 2，能够起发送多于一个网络链 4 的数据的作用，下文称之为分枝节点。可以沿至少一个网络链接受数据，并从一个分枝节点 2 指向其它网络链。在标准的形态中，数据可以从或沿任何连接到分枝节点去的网络链 4 接受或传送，这些分枝节点可以是，例如，一般用于包交换网络型式的路由器(routers)。

从分配服务器(DS)10 产生一个跨越网络的生成树，从该分配服务器要将一软件包 30 传送到一组必需接受软件的目标节点 14 上，并以网络中的节点 2 作为树中的分枝。在图 1 和 2 中以字母“T”代表目标节点，目标节点可以是树的叶子 16 或者可以是树中的分枝节点 2，在图 2 中表示这种达到目标的生成树。

树无需明确定义，同时可以，例如，隐含在每个分枝节点 2 上的路由器中。该分枝节点 2 发送数据是技术中众所周知的，这就是说，每个分枝节点保留一个对特殊节点给出目前可能的最短路线的路由表。这种结构是众所周知的，并且很好开发了算法以保留将提供产生从任何源到任何多个目的的生成树的适当路线的路由表。所使用的算法与方案，根据在所述及的网络的部分要使用的网络结构而变化，例如，有用于网桥连接 LANs 的特定的算法与方案。该算法中的一个在 IEEE801.1d 中描述，为在 Internet 基础系统上使用的这种结构在 REC1716 表示，它给出为在其 IP 地址的基础上有效发送 Internet 程序包的算法，路由器或等效的路由结构可以是静态的并手动更新，或者可以是基于在节点之间使用诸如 OSPF(Open Shortest Path First )Protocol 的发送方案传递的动态的，应该注意到只要路由机构保障，则在网络中的任意二节点之间，通常可取同样的路线。从一个节点到多个节点的传送通常将以生成树的型式出现。

需要向树中的每个分枝节点提供有关哪些节点需要特殊的简单的程序包 34 和组合的程序包 32，以构成一个要发送的总的软件包 30，在图 3A 中表示这种软件程序包的一个例子。根据此实施例，在 DS10 处作此决定同时可以包括在与软件包 30 一起发送的传递控制文件中。





根据第一实施例，当在一目标节点 14 上安装一程序包时，就保持那个安装的记录，因此未来的向那个目标节点的传送仅需包括所需要的。此记录可以在目标自身保持，由 DS10 转换的目标不论什么时候新的或更新的软件要被安装，或者可能由 DS 保留因此就不必须转换所有的目标，这两种方法的组合是可能的。

确定哪些目标节点需要哪些程序包所需的信息，因而既能够通过询问所有的目标节点有关什么样的程序已经被安装上来达到，又能通过保持跟踪安装在每个节点上的程序包来达到。当它们被安装时，如果由于任何理由，安装在特殊目标节点上的程序包不能被确定，可以发送整个软件包 30 到目标节点。

某些分枝节点 2 起在软件程序包 30 中再封装数据的作用将变得很明显，这些分枝节点在下文中将称为再封装节点 12。根据此实施例，这些节点设置能够存储整个软件程序包的存储媒体，但是，也设想一些实施例，在该实施例中由本发明的此实施例要求的再封装“在飞行中” (on the fly) 实现，而不需要足够的空间以存储软件包。

重要的是每个再封装节点 12 设置所有的目标节点 14，对该目标节点它将最终地传送数据，和被这些目标节点每一个所需要的程序包。这可以用许多方法容易地实施，二个例子如下：

1、数据与软件程序包 30 一起从 DS10 包括所有的目标节点 14 清单被传递，程序包要传递到该目标节点 14 上。任何接受程序包的分枝节点 2，检查其路由器，或等效的数据存储，以确定到所有传递程序包去的节点的最短路线，将沿着适合的分枝的每一个发送程序包的适当部分达到目标。分枝节点 2 整理沿每个分枝发送的目标节点清单，以便仅包括沿各自的分枝要达到的目标节点。

2、每个再封装节点 12 设置功能性以便确定负责传送从任何特殊节点发生的数据的所有目标，不需要在每个枝节点整理目标节点的清单，只要再封装节点知道发生的包群在哪里，如以上讨论的目标节点的清单可以存储在单一传送控制文件中。

以下例子表示根据本发明的第一实施例如何发送组合程序包，如图 3 所示，组合程序包 A 包括引用的简单程序包 B 和 C，程序包 B 的个别安



装是:

- 1) 运行 B 的安装前脚本
- 2) 安装 B 的文件
- 3) 运行 B 的安装后脚本

5 程序包 C 的个别安装是:

- 1) 运行 C 的安装前脚本
- 2) 安装 C 的文件
- 3) 运行 C 的安装后脚本

但是, 组合程序包 A 的安装是:

- 10 1) 运行 A 的安装前脚本
- 2) 运行 B 的安装前脚本
- 3) 安装 B 的文件
- 4) 运行 B 的安装后脚本
- 5) 运行 C 的安装前脚本
- 15 6) 安装 C 的文件
- 7) 运行 C 的安装后脚本
- 8) 运行 A 的安装后脚本

如果程序包 B 和程序包 C 已经个别地被安装在目标上(以分开的传送)它就意味着程序包 A 也被安装, 假如程序包 A 没有伴随其脚本。当一个程序包伴随其脚本时, 这就意味着脚本必须被执行由于认为它已被有效地安装。

对于一给定的传送, 软件程序包的存储信息在 DS 处集中, 并沿着生成树中的节点的系统发送到目标节点, DS 将程序包发送到树中的第一级分枝节点, 这些节点的每一个将它发送到下一级, 并且继续直至程序包已经发送到所有传送的目标节点, 用于发送程序包的中间节点也可以是目标节点。

不论什么时候, 后来的目标(包括并在要发送的一个以下的那些)仅需要程序包的内容的一部分再封装能够出现。在每一个目标处需要什么样的程序包的内容的确定在传送开始之前发生在 DS 处(并包括在传送控制文件中), 被后面的目标需要的程序包的存储信息组定义“新的程序包”。



在一个开放的程序包的情况下，可以仅仅选择引用的程序包并发送到需要的地方。在一个封闭的程序包的情况下，封闭的程序包以及它引用的所有程序包，直接地和间接地，必须被发送。

在以下的例子中，如图 4A、图 4B 和图 4C 所示，程序包包含二个组合程序包 a 和 c，以及三个简单的程序包 b, d 和 e，意向是将程序包 a 安装到目标 P、Q、R、S 和 T 上，程序包 a 是一个开放程序包，而认为程序包 c 是图 3B 中的开放情况和图 3C 所示的封闭情况二者。

在程序包 c 是开放的情况下(图 4B):

目标 P 具有安装的程序包 b 和 d 并需要程序包 e。

10 目标 Q 具有安装的程序 b 和 e 并需要程序包 d。也为目标 R、S 和 T 发送程序包 b 和 e。

目标 R 具有安装的程序包 b 和 d 并需程序包 e。

目标 S 上有安装的程序包 c(de)并需要程序包 b。

目标 T 具有安装的程序包 d 和 e 并需要程序包 b。

15 在程序包 C 是封闭的情况下(图 3C):

目标 P 具有安装的程序包 b 和 d 并需要程序包 c(de)。

目标 Q 具有安装的程序包 b 和 e 并需要程序包 c(de)。也为目标 S 和 T 发送程序包 b。

目标 R 具有安装的程序包 b 和 d 并需要程序包 c(de)。

20 目标 S 具有安装的程序包 c(de)并需要程序包 b。

目标 T 具有安装的程序包 d 和 e 并需要程序包 b 和 c(de)。

再封装构成软件程序包 30 的程序包的实现方法将在很大程度上根据所使用的方案和结构而变化，但可以看出这种再封装非常明确地实行使用从传送控制文件来的资料，因为对该资料需要程序包。例如，能以对应不同程序包的数组 (dimension) 以及对应不同分枝的数组来建立布尔列阵，在此列阵中的所有单元初始设置到 FALSE(不成立)。对再封装节点提供的程序通过目标节点的清单检测，当它找到一个对其将沿某个链传送的节点时，它设置在与第一数组中的链相关的列阵中的和由第二数组中的目标节点所需要的程序包中的列阵单元为 TRUE(成立)。一旦由程序认可所有的  
30 目标节点，沿每个链需要的软件包的内容将由与链相关的设置为 TRUE(成



立)的列阵中的存储单元来定义，图 5 表示这种型式的为再封装图 4C 中的节点 Q 的一个列阵的例子。

在本发明的第二实施例中，每个构成整个程序包的简单的和组合的程序包无需作为单一的连接的程序包被传送，但是每一个可以分开地与其自身的传送控制数据一起传送，以确保仅仅传送到选择的节点。构成整个程序包的每个程序包将有效地多点传送到适当的节点。如果一个程序包是封闭的组合程序包的一部分，则程序包可以设置指示目标节点的附件而不安装程序包直到它们接受原始的组合程序包。使用这种可选择的方法，将不必要执行再封装，并使用类似的整个带宽。当然，对每个发送的程序包可以根据其目标节点使用不同的生成树，这由图 6A 和图 6B 举例说明。在图 6A 中，在整个第一生成树上遍及链 101、103 和 104 发送程序包 B。在图 6B 中，在整个第二生成树上遍及不同的三个链 102、105 和 106 发送程序包 C。如果对程序包 A 和 B 二者使用相同的生成树，一个程序包就必须遍及 4 个链发送，例如，使用图 6A 的树，则必须遍及链 101、102、104 和 106 来传送程序包 C。

尽管已经表示并描述了本发明的最佳实施例，根据那些常规技术应该理解，在不偏离本发明在其更广的方面可作变化与修改，本发明的各种特征，在以下权利要求中陈述。







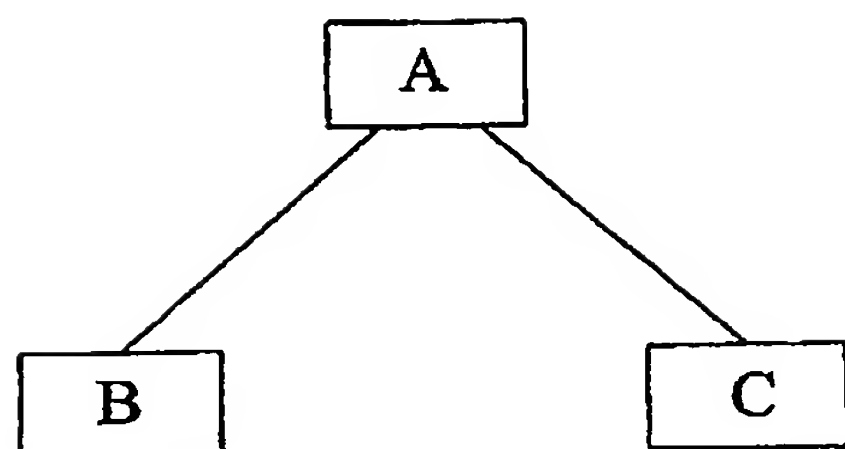


图3

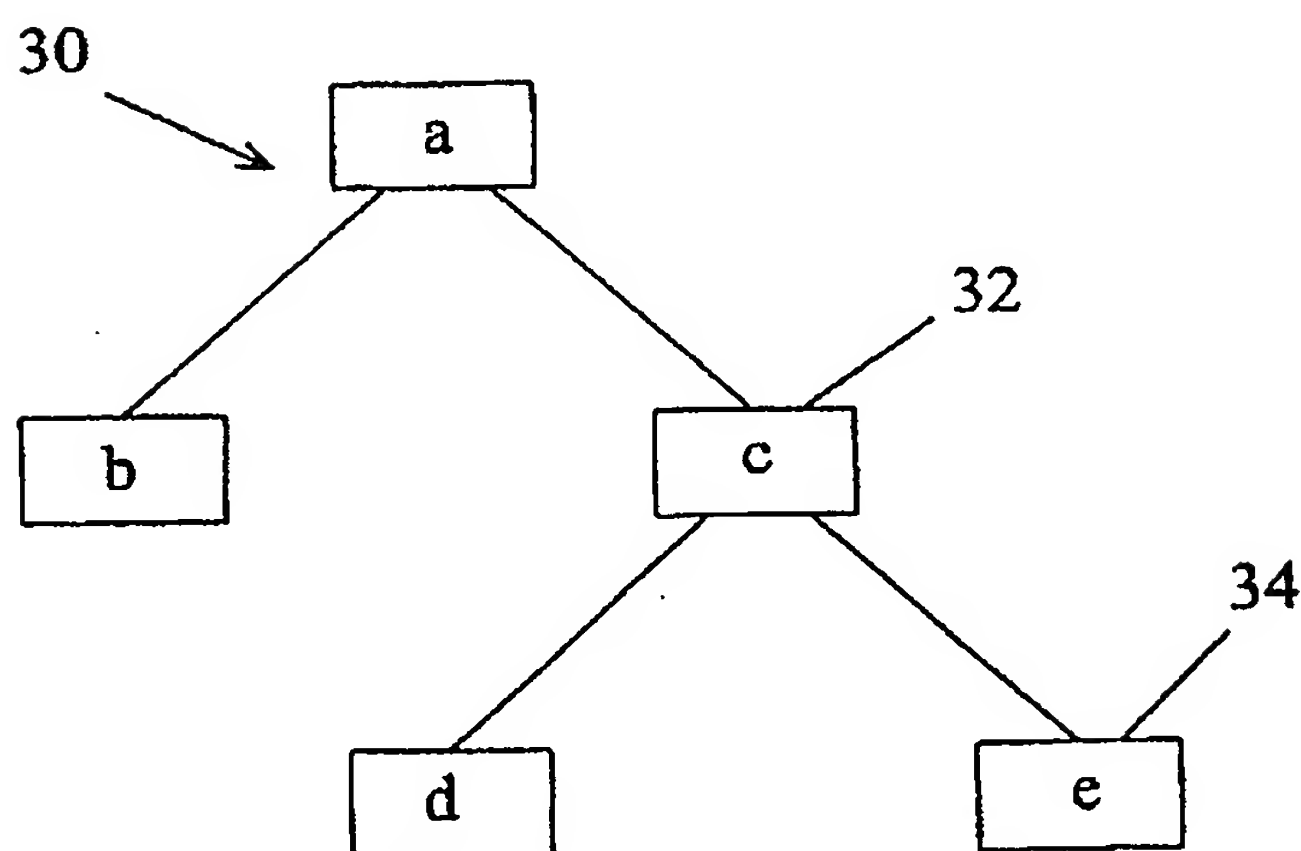


图4A

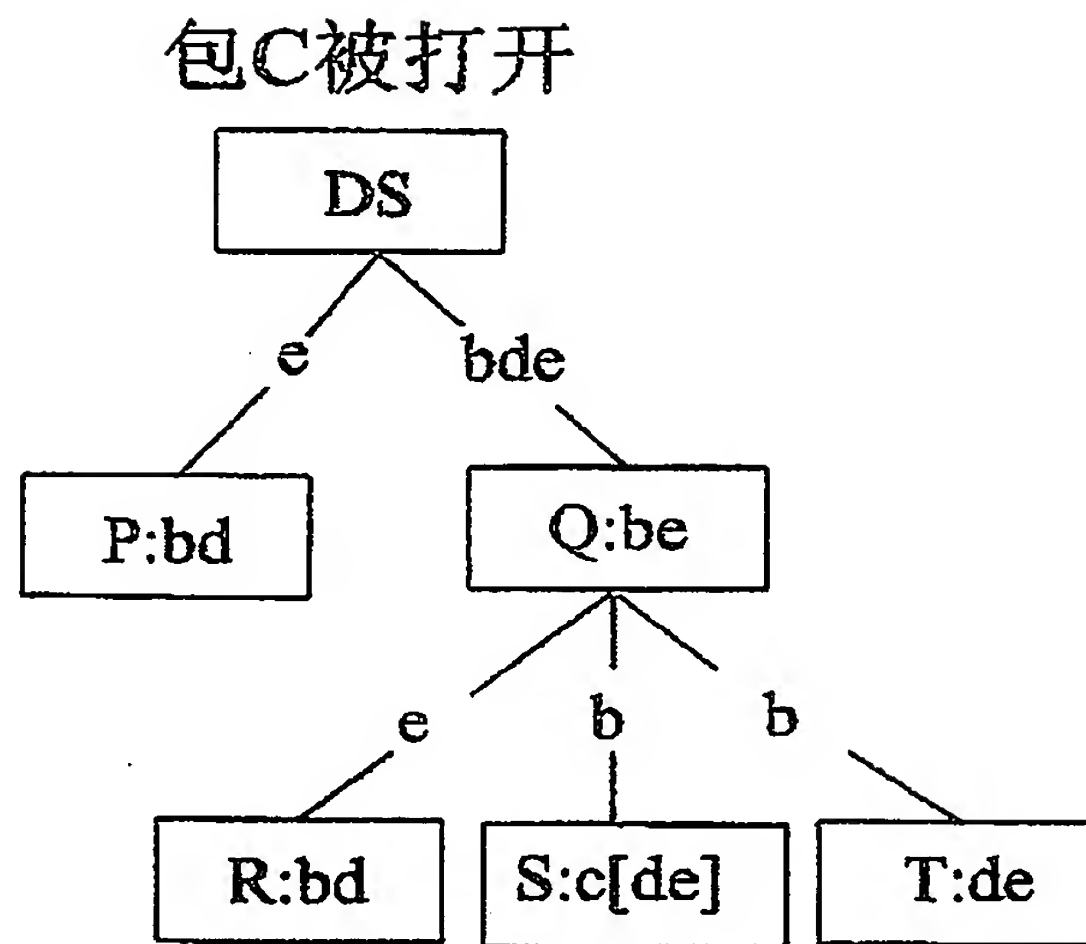


图4B

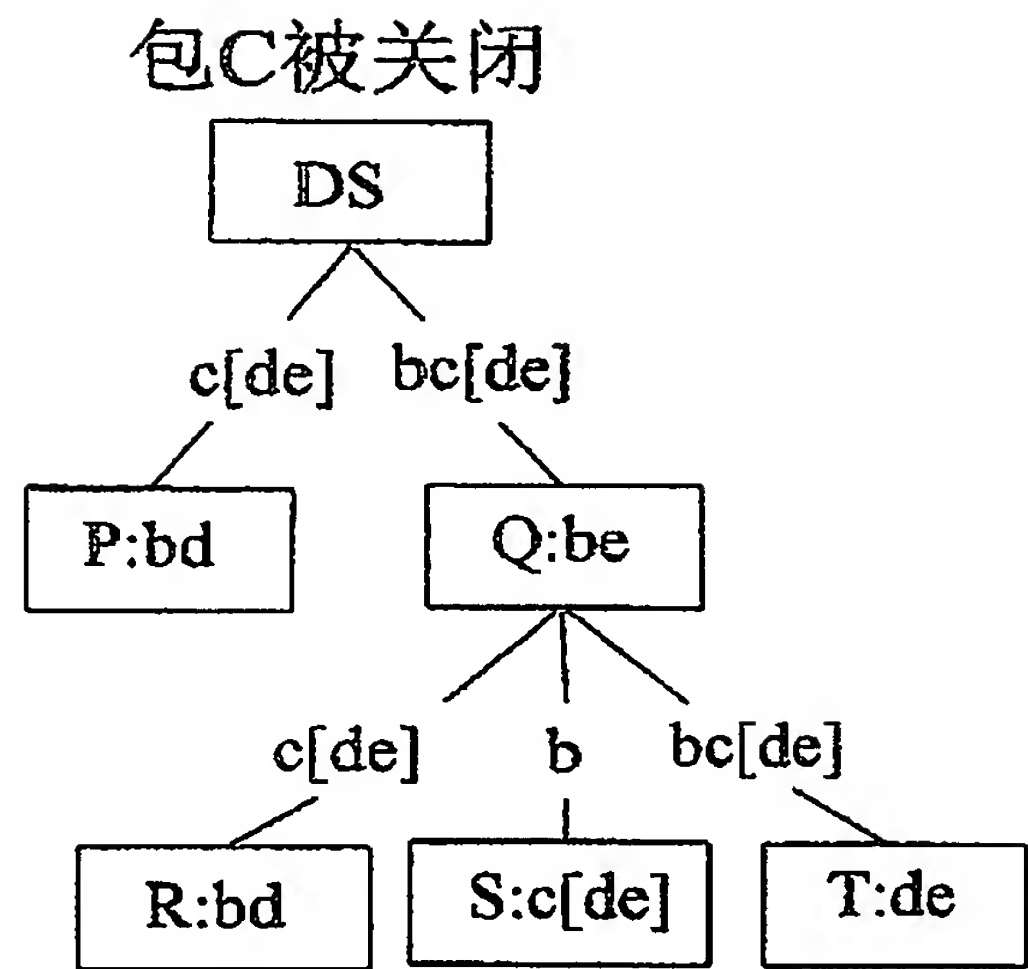


图4C

	b	c	d	e
1	F	T	T	T
2	T	F	F	F
3	T	T	T	T

图5



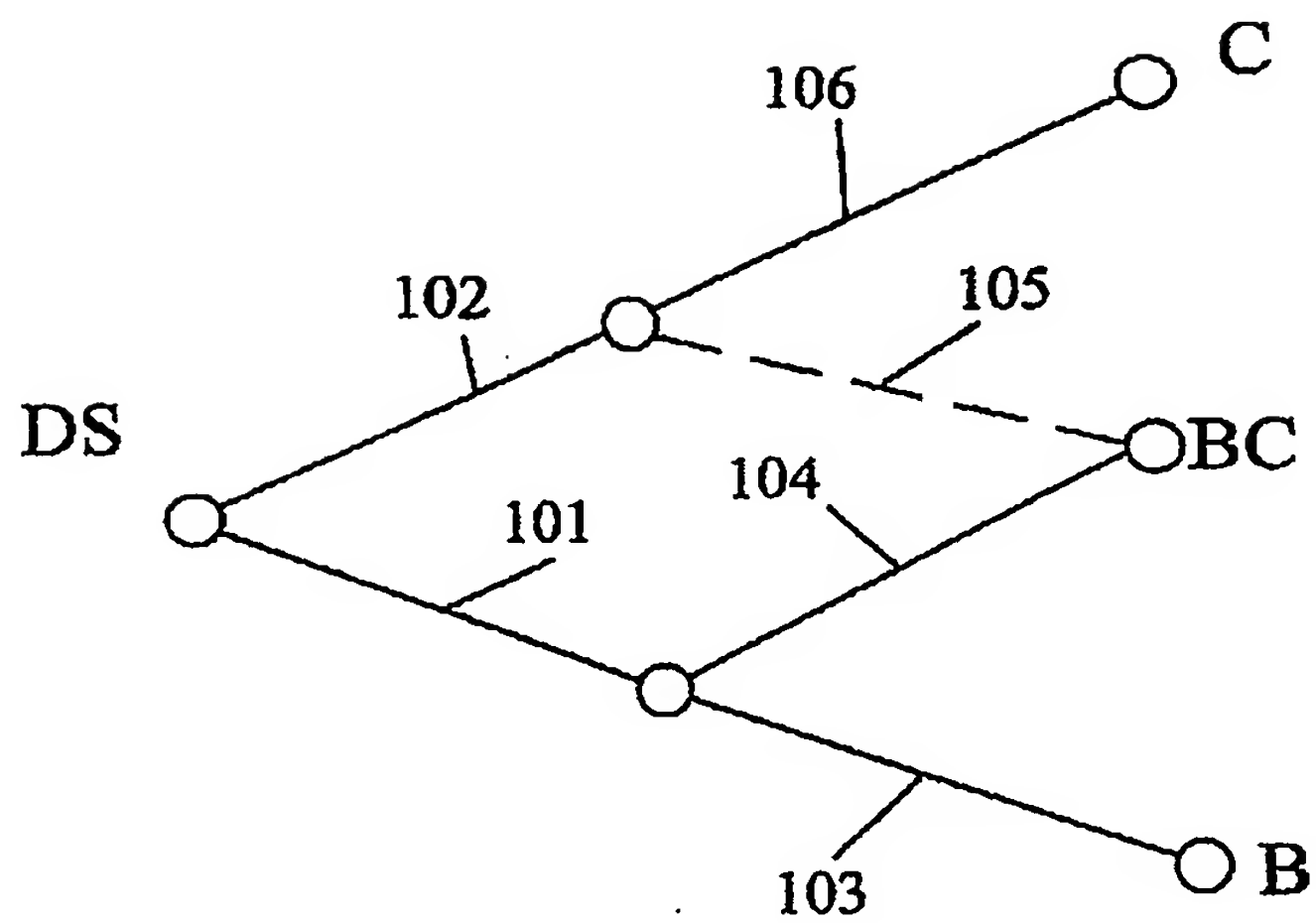


图6A

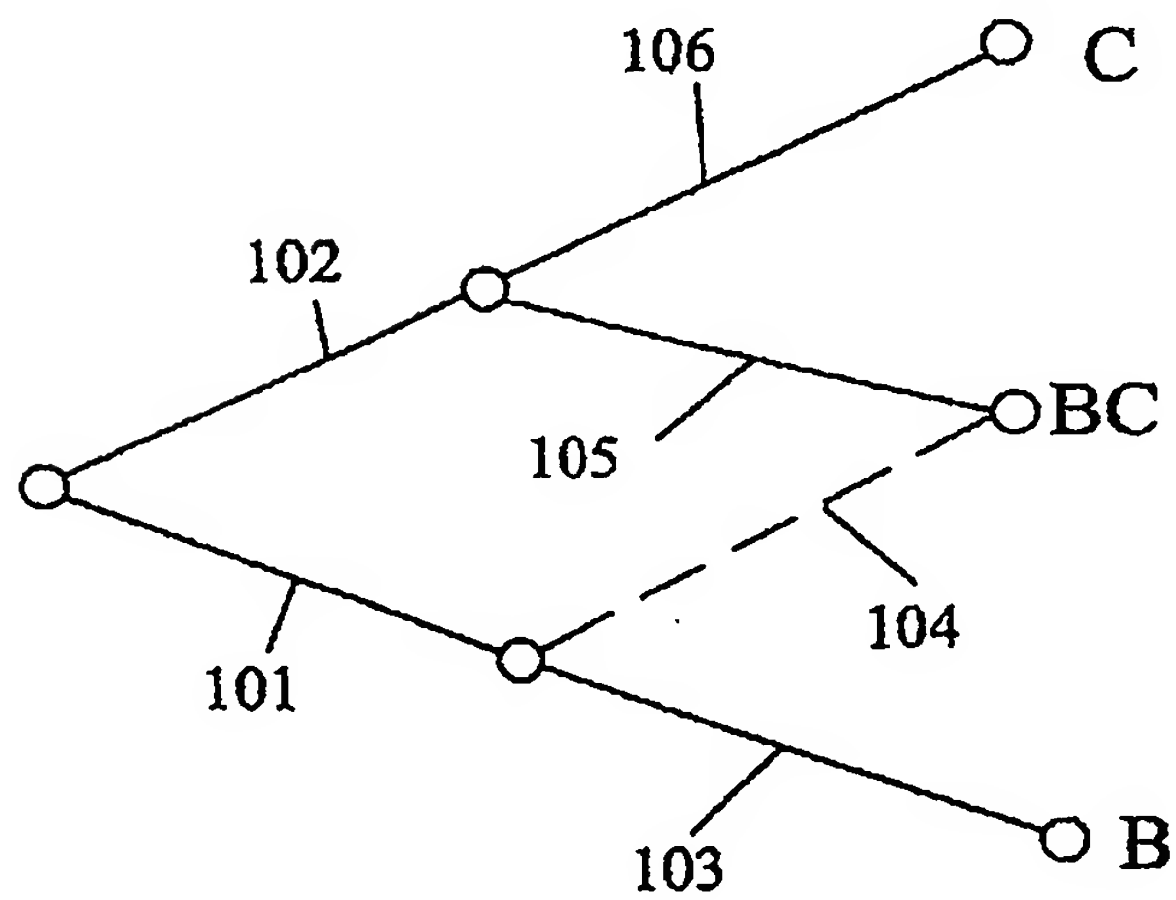


图6B